

Supplementary Material of the BMVC 2021 Paper: FFNB: Forgetting-Free Neural Blocks for Deep Continual Learning

This supplementary material includes the following items

- Incremental learning algorithms 1 and 2 (as discussed in section 3) of the paper.
- Detailed proof of proposition 1.
- Experiments showing the impact of the number of FFNB-layers on SBU and FPHA (tables 8 and 9).
- Experiments showing the impact of the band-size allowed to each task in the feature maps of FFNB on SBU and FPHA (tables 10 and 11); see also section 4.1 in the paper.
- Experiments showing the impact of pretraining the backbone network + FFNB fine-tuning on SBU and FPHA (tables 12 and 13); see also section 4.2 in the paper. Extra experiments are shown (in table 24) w.r.t. increasing sizes of pretraining data.
- Experiments showing a comparison between the aggregated “one-vs-one” classifiers w.r.t. the usual “one-vs-all” classifiers (tables 14 and 15); see also section 3.2.3.
- Analysis of different factors intervening in the bound (in Eq. 2): these factors correspond to dimension, weight decay regularization, and activation functions (tables 16, 17, 18, 19, 20 and 21).
- Experiments showing the impact of the batch-normalization with and without our covariance normalization on SBU and FPHA (tables 22 and 23) + a detailed justification about the performances in the underlying caption.
- And, experiments on CIFAR100, including comparisons w.r.t. the closely related work (tables 25, 26, 27 and 28).

Algorithms

Algorithm 1: Incremental learning

Input: Sequential tasks $\mathcal{T}_1, \dots, \mathcal{T}_T$.

Output: Trained network parameters $\{\mathbf{W}_{\ell,t}\}_{t,\ell}$.

for $t := 1$ **to** T **do**

repeat

Backward: back-propagate $\frac{\partial E}{\partial \Psi_{L+1}}$ and get $\frac{\partial E}{\partial \mathbf{W}_{\ell,t}}$;

$\mathbf{W}_{\ell,t} \leftarrow \mathbf{W}_{\ell,t} - \nu \frac{\partial E}{\partial \mathbf{W}_{\ell,t}}$; // being ν the learning rate

 Keep the parameters $\{\mathbf{W}_{\ell,r}\}_{r \neq t}$ of the other tasks unchanged;

Forward: update the outputs $\{\psi_\ell(\mathbf{X}_t)\}_\ell$ on the current task t ;

until convergence or max nbr of iterations reached;

Algorithm 2: Updated incremental learning

Input: Sequential tasks $\mathcal{T}_1, \dots, \mathcal{T}_T$.

Output: Trained network parameters $\{\mathbf{W}_{\ell,t}\}_{t,\ell}$.

for $t := 1$ **to** T **do**

 Set $\{\Phi^\ell\}_\ell$ layerwise using PCA on the previous task outputs $\{\psi_\ell(\mathbf{X}_\mathcal{P})\}_\ell$.

 Set $\{\alpha_{\ell,t}\}_\ell$ using Eq. (4) or (5).

repeat

Backward: back-propagate $\frac{\partial E}{\partial \Psi_{L+1}}$ and get $\{\frac{\partial E}{\partial \mathbf{W}_{\ell,t}}\}_\ell$;

$\frac{\partial E}{\partial \alpha_{\ell,t}} \leftarrow \frac{\partial E}{\partial \mathbf{W}_{\ell,t}} \frac{\partial \mathbf{W}_{\ell,t}}{\partial \alpha_{\ell,t}}$;

$\alpha_{\ell,t} \leftarrow \alpha_{\ell,t} - \nu \frac{\partial E}{\partial \alpha_{\ell,t}}, \forall \ell \in \{1, \dots, L\}$;

 Update $\mathbf{W}_{\ell,t}$ using Eq. (1) and keep $\{\mathbf{W}_{\ell,r}\}_{r \neq t}$ of the other tasks unchanged;

 Update $\{\mathbf{W}_{L,(t,r)}\}_{r \in \mathcal{P}}$ using Eq. (7) and keep the others unchanged;

Forward: update the outputs $\{\psi_\ell(\mathbf{X}_t)\}_\ell$ on the current task t ;

until convergence or max nbr of iterations reached;

Analysis on SBU and FPHA

In all the following tables, “null-space + heteroscedasticity + multi-task initialization” settings are used (following the ablation study in tables 3 and 4 in the paper). The number of FFNB feature layers and band-sizes are set to 3 (excepting particular settings in tables 8, 9 and 10, 11 respectively) and FFNB activations correspond to ReLU (excepting particular settings in tables 16 and 17). Pretraining and fine-tuning is always used (excepting particular settings in tables 12 and 13). Our aggregated “one-vs-one” classifiers are also used in all these tables (excepting particular settings in tables 14 and 15).

Note that accuracy when handling the first task in SBU is necessarily equal to 100% as the first task includes only one class, while in FPHA the first task includes 5 classes, so the accuracy is lower (see again captions of tables 1, 2, 3 and 4 in the paper).

# layers \ Tasks	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8
2 Layers	100.00	100.00	96.42	89.47	81.39	79.16	72.41	70.76
3 Layers	100.00	100.00	100.00	100.00	97.67	89.58	89.65	84.61
4 Layers	100.00	100.00	96.42	97.36	93.02	93.75	75.86	76.92

Table 8: Impact of the number of layers in the FFNB-features on the performances using SBU (in these experiments, $p = 45$).

# layers \ Tasks	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8	\mathcal{T}_9
3 Layers	68.25	58.73	64.76	63.70	62.84	61.24	64.96	66.01	67.47
4 Layers	53.96	44.44	35.75	40.15	46.13	44.96	46.78	45.50	46.95

Table 9: Impact of the number of layers in the FFNB-features on the performances using FPHA (in these experiments, $p = 75$).

Band size \ Tasks	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8
1	100.00	100.00	100.00	92.10	86.04	87.50	82.75	64.61
3	100.00	100.00	100.00	100.00	97.67	89.58	89.65	84.61
5	100.00	100.00	100.00	100.00	100.00	95.83	91.37	81.53
7	100.00	100.00	100.00	94.73	90.69	89.58	77.58	67.69
9	100.00	95.00	92.85	94.73	93.02	89.58	82.75	73.84

Table 10: Impact of band-size on the performances using SBU (again $p = 45$).

Band size \ Tasks									
	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8	\mathcal{T}_9
1	68.25	55.55	60.62	62.16	59.75	60.46	61.86	62.10	62.43
3	68.25	58.73	64.76	63.70	62.84	61.24	64.96	66.01	67.47
5	63.49	53.17	60.10	57.52	57.58	56.58	60.31	61.91	61.91
7	58.73	50.00	58.54	61.00	59.13	52.97	55.87	59.96	60.17
9	61.90	51.78	51.38	53.87	51.93	58.13	60.22	62.32	62.69

Table 11: Impact of band-size on the performances using FPHA ($p = 75$).

Pretraining	Fine-tuning	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8
\times	\times	100.00	100.00	97.36	95.34	93.75	87.93	83.07	83.07
\times	\checkmark	100.00	100.00	97.36	93.02	87.50	87.93	84.61	84.61
\checkmark	\times	100.00	100.00	100.00	97.36	90.69	87.50	87.93	83.07
\checkmark	\checkmark	100.00	100.00	100.00	100.00	97.67	89.58	89.65	84.61

Table 12: Impact of pretraining and fine-tuning on the performances using SBU (here $p = 45$).

Pretraining	Fine-tuning	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8	\mathcal{T}_9
\times	\times	63.15	59.52	59.58	58.30	58.82	57.10	60.75	62.89	63.13
\times	\checkmark	64.47	60.86	62.43	56.08	58.20	55.13	58.74	58.96	60.34
\checkmark	\times	68.25	59.52	61.65	61.77	59.13	58.39	60.31	61.52	63.30
\checkmark	\checkmark	68.25	58.73	64.76	63.70	62.84	61.24	64.96	66.01	67.47

Table 13: Impact of pretraining and fine-tuning on the performances using FPHA (here $p = 75$).

FFNB classifiers \ Tasks									
	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8	
“One-vs-all” classifiers	100.00	100.00	82.14	81.57	65.11	66.66	60.34	55.38	
Our aggregated “one-vs-one” classifiers	100.00	100.00	97.36	93.02	87.50	87.93	84.61	84.61	

Table 14: Impact of FFNB-classifiers on the performances using SBU (in these experiments, $p = 45$).

FFNB classifiers \ Tasks									
	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8	\mathcal{T}_9
“One-vs-all” classifiers	38.09	29.36	47.66	43.62	36.53	36.17	35.25	35.74	35.65
Our aggregated “one-vs-one” classifiers	68.25	58.73	64.76	63.70	62.84	61.24	64.96	66.01	67.47

Table 15: Impact of FFNB-classifiers on the performances using FPHA (in these experiments, $p = 75$).

FFNB-Activations \ Performances		
	CF Bound in Eq (2)	Accuracy (@final task \mathcal{T}_8)
Tanh	0.1581	72.30
Sigmoid	5.75×10^{-5}	76.92
ReLU	15.98×10^{-7}	84.61

Table 16: Impact of activations on the catastrophic forgetting (CF) bound and performances using SBU ($p = 45$).

FFNB-Activations \ Performances	CF Bound in Eq (2)	Accuracy (@final task \mathcal{T}_9)
Tanh	19.23	60.86
Sigmoid	18.52	60.86
ReLU	0.288	67.47

Table 17: Impact of activations on the CF bound and performances using FPHA ($p = 75$).

Dimensions \ Performances	CF Bound in Eq (2) ($\times 10^{-7}$)	Accuracy (@final task \mathcal{T}_8)
$p = 15$	25618.3	81.53
$p = 25$	437.39	83.07
$p = 35$	64.12	81.53
$p = 45$	15.98	84.61
$p = 50$	7.80	73.84
$p = 55$	2.54	69.23

Table 18: Impact of dimensions (p) on the CF bound and performances using SBU (with ReLU). From these results, small p leads to CF while large p to low dimensional and noisy null-space (and hence low generalization), so the best performances are obtained when sufficiently (but not very) large p -values are selected.

Dimensions \ Performances	CF Bound in Eq (2)	Accuracy (@final task \mathcal{T}_9)
$p = 15$	425.716411	63.82
$p = 30$	11.005551	61.56
$p = 45$	2.656760	63.30
$p = 60$	0.836268	64.69
$p = 75$	0.288356	67.47
$p = 90$	0.115226	60.69
$p = 105$	0.043147	60.00
$p = 120$	0.015686	57.21

Table 19: Impact of dimensions (p) on the CF bound and performances using FPHA (with ReLU). Again, from these results, small p leads to CF while large p to low dimensional and noisy null-space (and hence low generalization), so the best performances are obtained when sufficiently (but not very) large p -values are selected.

Weight decay coefficient \ Performances	CF Bound in Eq (2) ($\times 10^{-7}$)	Accuracy (@final task \mathcal{T}_8)
10^{-8}	15.98	84.61
10^{-7}	15.86	83.07
10^{-6}	14.89	81.53
10^{-5}	7.59	78.46
10^{-4}	6.59	78.46
10^{-3}	4.51	81.53
10^{-2}	2.10	76.92

Table 20: Impact of weight decay regularization on the CF bound and performances using SBU (with ReLU and $p = 45$).

Weight decay coefficient \ Performances	CF Bound in Eq (2)	Accuracy (@final task \mathcal{T}_9)
10^{-8}	0.288356	67.47
10^{-7}	0.281536	64.00
10^{-6}	0.238607	64.17
10^{-5}	0.170736	55.13
10^{-4}	0.134717	55.82
10^{-3}	0.116821	47.65
10^{-2}	0.139448	47.47

Table 21: Impact of weight decay regularization on the CF bound and performances using FPHA (with ReLU and $p = 75$).

Batch-norm	Heteroscedasticity	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8
\times	\times	100.00	100.00	100.00	97.36	90.69	87.50	82.75	75.38
\times	\checkmark	100.00	100.00	100.00	100.00	97.67	89.58	89.65	84.61
\checkmark	\times	100.00	85.00	67.85	55.26	41.86	33.33	32.75	29.23
\checkmark	\checkmark	100.00	100.00	96.42	100.00	95.34	95.83	93.10	83.07

Table 22: Impact of batch-norm (BN), with and w/o our class-wise covariance normalization, on SBU (here $p = 45$). The reason why BN is degrading performances is not intrinsically related to the BN itself (which is known to be effective in the general multi-task setting), but due to the incremental setting (i.e., due to the interference introduced by the BN on the previous tasks; put differently, the feature maps of the FFNB network on the previous tasks are no longer guaranteed to belong to the residual space when BN is applied).

Batch-norm	Heteroscedasticity	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8	\mathcal{T}_9
\times	\times	76.19	67.46	64.76	62.93	62.84	61.24	59.42	58.00	54.95
\times	\checkmark	68.25	58.73	64.76	63.70	62.84	61.24	64.96	66.01	67.47
\checkmark	\times	52.38	45.23	34.71	32.81	36.22	39.53	39.24	45.31	49.21
\checkmark	\checkmark	74.60	57.93	33.16	44.01	48.60	53.74	58.09	58.98	57.39

Table 23: Impact of batch normalization (with and w/o covariance normalization) on FPHA (here $p = 75$). We observe a similar behavior as SBU (see caption of the previous table).

Configuration	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8	\mathcal{T}_9
no-pretraining	63.15	59.52	59.58	58.30	58.82	57.10	60.75	62.89	63.13
pretraining (25% of pretraining data)	68.25	49.20	55.95	58.30	63.46	61.49	62.97	62.69	63.47
pretraining (50% of pretraining data)	57.14	56.34	63.21	61.77	60.68	62.27	63.85	63.28	63.82
pretraining (100% of pretraining data)	68.25	58.73	64.76	63.70	62.84	61.24	64.96	66.01	67.47

Table 24: Impact of backbone pretraining on the performances using FPHA (here $p = 75$). Results shown in this table provide an idea about the behavior of our FFNB w.r.t. increasing pretraining sets and also w.r.t. no-pretraining of the backbone. In spite of no-pretraining, FFNB (also endowed with feature map layers) is able to adapt the features to the new incremental tasks prior to achieve classification. This is possible thanks to the new dynamic parameters of the current task which are trained in the null-space of the previous tasks, and this mitigates CF.

Evaluation on CIFAR100 and SOTA Comparison

In all the following tables, “null-space + heteroscedasticity + multi-task initialization” settings are used. On CIFAR100, the band-size=1, size of minibatch=32, optimizer=SGD, learning rate fixed to 10e-3 and neither weight decay nor momentum are used.

Test classes \ Tasks	\mathcal{T}_0	\mathcal{T}_1	\mathcal{T}_2	Average Incremental Acc.
Top-50 classes	83.66	76.18	68.40	76.08
50 – 75		60.96	51.52	56.24
75 – 100	–	–	60.68	60.68
Average Task Acc.	83.66	71.11	62.25	72.34

Table 25: Results on *CIFAR100-B50-S2*. As suggested by the standard evaluation protocol, the first 50 classes ([1-50]) are used to pretrain the “EfficientNet” backbone, while the remaining 50 classes ([51-100]) are used for incremental task learning. Here *B50* stands for these 50 pretraining classes and *S2* for tasks \mathcal{T}_1 and \mathcal{T}_2 which are learned incrementally (here \mathcal{T}_1 corresponds to classes [51-75] and \mathcal{T}_2 to [76-100] while \mathcal{T}_0 is the pretraining task involving classes [1-50]). *Average Incremental Acc* is proposed in iCaRL [RKSL17] which is averaged across tasks. The symbol “–” stands for “accuracy not available” as classes are incrementally visited so training+test data, belonging to the subsequent tasks, are obviously not available beforehand.

Test classes \ Tasks	\mathcal{T}_0	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	Average Incremental Acc.
Top-50 classes	83.66	79.26	74.74	70.76	64.76	58.18	71.89
50 – 60		76.80	71.30	67.00	58.20	50.30	64.72
60 – 70			63.10	57.10	50.00	41.50	52.93
70 – 80	–	–	–	69.10	61.10	52.60	60.73
80 – 90					75.50	69.00	72.25
90 – 100					–	70.90	70.90
Average Task Acc.	83.66	78.55	72.59	68.38	63.18	57.52	70.70

Table 26: Results on *CIFAR100-B50-S5*. The caption of this table is similar to table 25 excepting the number of tasks which is now equal to 5.

We use EfficientNet [TL19] as our backbone which is state-of-the-art feature extractor architecture. Comparison shown in table 28 are w.r.t. the following related work

- [HPL+19] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 831–839, 2019.
- [HTM+21] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. arXiv preprint arXiv:2103.01737, 2021.
- [RKSL17] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2001–2010, 2017.

Tasks Test classes	\mathcal{T}_0	\mathcal{T}_1	\mathcal{T}_2	\mathcal{T}_3	\mathcal{T}_4	\mathcal{T}_5	\mathcal{T}_6	\mathcal{T}_7	\mathcal{T}_8	\mathcal{T}_9	\mathcal{T}_{10}	Average Incremental Acc.
Top-50 classes	83.66	80.98	78.02	73.90	71.32	68.20	64.82	61.42	57.52	52.60	49.42	67.44
50 – 55		79.20	71.60	70.40	65.80	58.00	56.60	43.80	43.60	34.00	26.00	54.90
55 – 60			89.40	84.00	74.20	71.40	68.60	63.80	61.80	60.20	57.00	70.04
60 – 65				62.80	58.00	56.40	46.00	43.40	35.20	31.00	24.20	44.63
65 – 70					73.60	73.20	66.40	62.80	55.40	50.80	43.00	60.74
70 – 75						80.60	76.40	68.00	59.40	47.80	41.60	62.30
75 – 80	–	–					80.80	80.20	71.20	67.60	62.40	72.44
80 – 85			–	–				87.40	73.00	62.80	57.20	70.10
85 – 90					–	–			85.00	83.40	79.00	82.47
90 – 95							–	–		82.60	74.60	78.60
95 – 100										–	80.40	80.40
Average Task Acc.	83.66	80.82	78.43	73.55	70.34	68.11	65.19	62.56	58.88	55.06	51.98	68.05

Table 27: Results on *CIFAR100-B50-S10*. The caption of this table is similar to table 25 excepting the number of tasks which is now equal to 10.

Methods	2 tasks (S2)		5 tasks (S5)		10 tasks (S10)	
	#Params (M)	Avg. Acc.	#Params (M)	Avg. Acc.	#Params (M)	Avg. Acc.
Upper Bound [YXH21]	11.2	67.38 / 72.22	11.2	79.89	11.2	79.91
iCaRL[RKSL17,YXH21]	11.2	71.33	11.2	65.06	11.2	58.59
UCIR [HPL+19,YXH21]	11.2	67.21	11.2	64.28	11.2	59.92
BiC [WCW+19,YXH21]	11.2	72.47	11.2	66.62	11.2	60.25
WA [ZXG+20,YXH21]	11.2	71.43	11.2	64.01	11.2	57.86
PoDNet [YXH21]	11.2	71.30	11.2	67.25 (64.83)	11.2	64.04 (63.19)
DDE (UCIR R20) [HTM+21]	-	-	11.2	65.27	11.2	62.36
DDE (PoDNet R20) [HTM+21]	-	-	11.2	65.42	11.2	64.12
DER(w/o P) [YXH21]	22.4	74.61	39.2	73.21	67.2	72.81
DER(P) [YXH21]	3.90	74.57	6.13	72.60	8.79	72.45
Ours	5.8	72.34	5.8	70.70	5.8	68.05

Table 28: Results on *CIFAR100-B50* (modified from Table 2 in DER [YXH21] where numbers in blue refer to the results tested by the re-implementation in DER [YXH21] and numbers in parentheses refer to the results reported in the original papers).

- [TL19] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning, pages 6105–6114. PMLR, 2019.
- [WCW+19] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 374–382, 2019.
- [YXH21] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. arXiv preprint arXiv:2103.16788, 2021.
- [ZXG+20] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13208–13217, 2020.

Proposition 1

Proposition 1 Let $g : \mathbb{R} \rightarrow \mathbb{R}$ be a L -Lipschitz continuous activation (with $L \leq 1$). Any η -step descent (update) of $\mathbf{W}_{\ell,t}$ in $\mathcal{N}_S(\Psi_\ell(\mathbf{X}_{\mathcal{P}}))$ using (1) satisfies $\forall r \in \mathcal{P}$

$$\|\psi_\ell^\eta(\mathbf{X}_r) - \psi_\ell^0(\mathbf{X}_r)\|_F^2 \leq B$$

$$\text{with} \quad B = \sum_{\tau=1}^{\eta} \sum_{k=0}^{\ell-1} (\|\alpha_{\ell-k,t}^\tau\|_F^2 \|\beta_{\ell-k-1,r}^\tau\|_F^2 + \|\alpha_{\ell-k,t}^{\tau-1}\|_F^2 \|\beta_{\ell-k-1,r}^{\tau-1}\|_F^2) \cdot \prod_{k'=0}^{k-1} \|\mathbf{W}_{\ell-k',\mathcal{P}}^\tau\|_F^2, \quad (10)$$

being $\psi_\ell^0(\mathbf{X}_r)$ (resp. $\psi_\ell^{\eta-1}(\mathbf{X}_r)$) the map before the start (resp. the end) of the iterative update (descent on current task \mathcal{T}_t), $\beta_{\ell,r}^\tau$ the projection of $\psi_\ell^\tau(\mathbf{X}_r)$ onto $\mathcal{N}_S(\Psi_\ell(\mathbf{X}_{\mathcal{P}}))$ at any iteration τ , $\{\mathbf{W}_{\ell,r}^\tau\}_\ell$ the network parameters at τ , and $\|\cdot\|_F$ the Frobenius norm.

Proof of Proposition 1

At any iteration τ of the descent, one may write $\forall r \in \mathcal{P}$

$$\begin{aligned} \|\psi_\ell^\tau(\mathbf{X}_r) - \psi_\ell^{\tau-1}(\mathbf{X}_r)\|_F^2 &= \|g(\mathbf{W}_\ell^\tau \psi_{\ell-1}^\tau(\mathbf{X}_r)) - g(\mathbf{W}_\ell^{\tau-1} \psi_{\ell-1}^{\tau-1}(\mathbf{X}_r))\|_F^2 \\ &\leq \|\mathbf{W}_\ell^\tau \psi_{\ell-1}^\tau(\mathbf{X}_r) - \mathbf{W}_\ell^{\tau-1} \psi_{\ell-1}^{\tau-1}(\mathbf{X}_r)\|_F^2 \quad (g \text{ } L\text{-Lipschitzian with } L \leq 1) \\ &= \|\mathbf{W}_{\ell,t}^\tau \psi_{\ell-1}^\tau(\mathbf{X}_r) - \mathbf{W}_{\ell,t}^{\tau-1} \psi_{\ell-1}^{\tau-1}(\mathbf{X}_r)\|_F^2 \\ &\quad + \|\mathbf{W}_{\ell,\mathcal{P}}^\tau \psi_{\ell-1}^\tau(\mathbf{X}_r) - \mathbf{W}_{\ell,\mathcal{P}}^{\tau-1} \psi_{\ell-1}^{\tau-1}(\mathbf{X}_r)\|_F^2 \quad (\mathbf{W}_{\ell,\mathcal{P}}^{\tau-1} = \mathbf{W}_{\ell,\mathcal{P}}^\tau, \forall \tau) \\ &\leq \|\mathbf{W}_{\ell,t}^\tau \psi_{\ell-1}^\tau(\mathbf{X}_r) - \mathbf{W}_{\ell,t}^{\tau-1} \psi_{\ell-1}^{\tau-1}(\mathbf{X}_r)\|_F^2 \\ &\quad + \|\mathbf{W}_{\ell,\mathcal{P}}^\tau\|_F^2 \|\psi_{\ell-1}^\tau(\mathbf{X}_r) - \psi_{\ell-1}^{\tau-1}(\mathbf{X}_r)\|_F^2 \quad (\text{Cauchy Schwarz}) \\ &= \|(\alpha_{\ell,t}^\tau)^\top \Phi^\top \psi_{\ell-1}^\tau(\mathbf{X}_r) - (\alpha_{\ell,t}^{\tau-1})^\top \Phi^\top \psi_{\ell-1}^{\tau-1}(\mathbf{X}_r)\|_F^2 \quad (\text{Eq. 1 in paper}) \\ &\quad + \|\mathbf{W}_{\ell,\mathcal{P}}^\tau\|_F^2 \|\psi_{\ell-1}^\tau(\mathbf{X}_r) - \psi_{\ell-1}^{\tau-1}(\mathbf{X}_r)\|_F^2 \\ &= \|(\alpha_{\ell,t}^\tau)^\top \Phi^\top \Phi \beta_{\ell-1,r}^\tau - (\alpha_{\ell,t}^{\tau-1})^\top \Phi^\top \Phi \beta_{\ell-1,r}^{\tau-1}\|_F^2, \quad (\psi_{\ell-1}^\tau(\mathbf{X}_r) = \Phi \beta_{\ell-1,r}^\tau) \\ &\quad + \|\mathbf{W}_{\ell,\mathcal{P}}^\tau\|_F^2 \|\psi_{\ell-1}^\tau(\mathbf{X}_r) - \psi_{\ell-1}^{\tau-1}(\mathbf{X}_r)\|_F^2 \\ &= \|(\alpha_{\ell,t}^\tau)^\top \beta_{\ell-1,r}^\tau - (\alpha_{\ell,t}^{\tau-1})^\top \beta_{\ell-1,r}^{\tau-1}\|_F^2 \quad (\{\Phi_d\}_d \text{ orthonormal}) \\ &\quad + \|\mathbf{W}_{\ell,\mathcal{P}}^\tau\|_F^2 \|\psi_{\ell-1}^\tau(\mathbf{X}_r) - \psi_{\ell-1}^{\tau-1}(\mathbf{X}_r)\|_F^2 \\ &\leq \|\alpha_{\ell,t}^\tau\|_F^2 \|\beta_{\ell-1,r}^\tau\|_F^2 + \|\alpha_{\ell,t}^{\tau-1}\|_F^2 \|\beta_{\ell-1,r}^{\tau-1}\|_F^2 \quad (\text{Cauchy Schwarz}) \\ &\quad + \|\mathbf{W}_{\ell,\mathcal{P}}^\tau\|_F^2 \|\psi_{\ell-1}^\tau(\mathbf{X}_r) - \psi_{\ell-1}^{\tau-1}(\mathbf{X}_r)\|_F^2. \end{aligned}$$

Combining the above inequality using recursion

$$\begin{aligned} \|\psi_\ell^\tau(\mathbf{X}_r) - \psi_\ell^{\tau-1}(\mathbf{X}_r)\|_F^2 &\leq \sum_{k=0}^{\ell-1} (\|\alpha_{\ell-k,t}^\tau\|_F^2 \|\beta_{\ell-k-1,r}^\tau\|_F^2 + \|\alpha_{\ell-k,t}^{\tau-1}\|_F^2 \|\beta_{\ell-k-1,r}^{\tau-1}\|_F^2) \cdot \prod_{k'=0}^{k-1} \|\mathbf{W}_{\ell-k',\mathcal{P}}^\tau\|_F^2 \\ &\quad + \prod_{k'=0}^{\ell-1} \|\mathbf{W}_{\ell-k',\mathcal{P}}^\tau\|_F^2 \|\psi_0^\tau(\mathbf{X}_r) - \psi_0^{\tau-1}(\mathbf{X}_r)\|_F^2 \\ &= \sum_{k=0}^{\ell-1} (\|\alpha_{\ell-k,t}^\tau\|_F^2 \|\beta_{\ell-k-1,r}^\tau\|_F^2 + \|\alpha_{\ell-k,t}^{\tau-1}\|_F^2 \|\beta_{\ell-k-1,r}^{\tau-1}\|_F^2) \cdot \prod_{k'=0}^{k-1} \|\mathbf{W}_{\ell-k',\mathcal{P}}^\tau\|_F^2, \end{aligned}$$

$\|\psi_0^\tau(\mathbf{X}_r) - \psi_0^{\tau-1}(\mathbf{X}_r)\|_F^2 = \|\mathbf{X}_r - \mathbf{X}_r\|_F^2 = 0$ as the parameters of the convolutional layers of the whole network f are initially pretrained and fixed, so any incremental training of f maintains $\{\psi_0(\mathbf{X}_r)\}_r$ unchanged. Considering η the max number of epochs when training the parameters of the t^{th} task, one may write

$$\begin{aligned} \|\psi_\ell^\eta(\mathbf{X}_r) - \psi_\ell^0(\mathbf{X}_r)\|_F^2 &= \|\psi_\ell^\eta(\mathbf{X}_r) - \sum_{\tau=1}^{\eta-1} \psi_\ell^\tau(\mathbf{X}_r) + \sum_{\tau=1}^{\eta-1} \psi_\ell^\tau(\mathbf{X}_r) - \psi_\ell^0(\mathbf{X}_r)\|_F^2 \\ &\leq \sum_{\tau=1}^{\eta} \|\psi_\ell^\tau(\mathbf{X}_r) - \psi_\ell^{\tau-1}(\mathbf{X}_r)\|_F^2. \end{aligned}$$

Combining the two above inequalities, it follows

$$\begin{aligned} \|\psi_\ell^\eta(\mathbf{X}_r) - \psi_\ell^0(\mathbf{X}_r)\|_F^2 &\leq \sum_{\tau=1}^{\eta} \sum_{k=0}^{\ell-1} (\|\alpha_{\ell-k,t}^\tau\|_F^2 \cdot \|\beta_{\ell-k-1,r}^\tau\|_F^2 + \|\alpha_{\ell-k,t}^{\tau-1}\|_F^2 \cdot \|\beta_{\ell-k-1,r}^{\tau-1}\|_F^2) \\ &\quad \cdot \prod_{k'=0}^{k-1} \|\mathbf{W}_{\ell-k',\mathcal{P}}^\tau\|_F^2. \end{aligned}$$

■